



Karnik, A. A., Mayol-Cuevas, W., & Subramanian, S. (2012). MUSTARD: a multi user see through AR display. In *Proceedings of the 2012 ACM annual conference on Human Factors in Computing Systems* (pp. 2541-2550). (CHI '12). Association for Computing Machinery (ACM). <https://doi.org/10.1145/2207676.2208641>

Peer reviewed version

Link to published version (if available):
[10.1145/2207676.2208641](https://doi.org/10.1145/2207676.2208641)

[Link to publication record in Explore Bristol Research](#)
PDF-document

© ACM, 2012. This is the author's version of the work. It is posted here by permission of ACM for your personal use. Not for redistribution. The definitive version was published in CHI 2012, May 5-10, 2012, Austin, TX, USA.
<http://dl.acm.org/citation.cfm?doid=2207676.2208641>

University of Bristol - Explore Bristol Research

General rights

This document is made available in accordance with publisher policies. Please cite only the published version using the reference above. Full terms of use are available:
<http://www.bristol.ac.uk/red/research-policy/pure/user-guides/ebr-terms/>

MUSTARD: A Multi User See Through AR Display

Abhijit Karnik, Walterio Mayol-Cuevas, Sriram Subramanian

Department of Computer Science,
University of Bristol, UK
karnik, wmayol, sriram @cs.bris.ac.uk

ABSTRACT

We present MUSTARD, a multi-user dynamic random hole see-through display, capable of delivering viewer dependent information for objects behind a glass cabinet. Multiple viewers are allowed to observe both the physical object(s) being augmented and their location dependent annotations at the same time. The system consists of two liquid-crystal (LC) panels within which physical objects can be placed. The back LC panel serves as a dynamic mask while the front panel serves as the data. We first describe the principle of MUSTARD and then examine various functions that can be used to minimize crosstalk between multiple viewer positions. We compare different conflict management strategies using PSNR and the quality mean opinion score of HDR-VDP2. Finally, through a user-study we show that users can clearly identify images and objects even when the images are shown with strong conflicting regions; demonstrating that our system works even in the most extreme of circumstances.

Author Keywords

Display; Random hole display; see-through; multi-view; multi-user; Augmented Reality.

ACM Classification Keywords

H5.2 [Information interfaces and presentation]: User Interfaces - Graphical user interfaces;

General Terms

Design, Human Factors

INTRODUCTION

There are many settings in which users are expected to experience physical objects through a glass cabinet. Such settings exist in museums, shops and vending machines. In a museum it is common for users to view delicate real objects placed behind a protective glass without direct access to handle them. In the case of shopping windows or vending machines where some transaction may be required to gain access to the physical object they also offer a means of finding out more about the item prior to purchase. Retailers have already trialed simple systems aiming at

combining real objects and interactive information. For e.g., Polo Ralph Lauren unveiled an interactive shopping window (engadget, Aug. 10, 2006), in which shoppers could view clothing and make purchases via an interactive touch screen.

However these and similar systems are limited to being tailored to work for one user at a time and information is not spatially situated. Thus the experience is closer to navigating a web catalog than properly augmenting the object behind the glass. The most beneficial aspect of an augmented reality see-through glass in a museum or retail setting would be its ability to support multiple users with distinct views. Shop windows are often explicitly designed to encourage multiple users to engage with their content.

The two opposing challenges are: a) delivering unimpeded visibility of the physical object behind a semi-transparent layer and b) simultaneously providing viewer-dependent information for multiple viewers.

Although Parallax barriers and lenticular arrays may be used to provide viewer-dependent information to multiple viewers they have not been shown to support the see-through feature described above. Especially, lenticular arrays distort views of the objects placed behind them limiting their use as a multi-view see-through display.

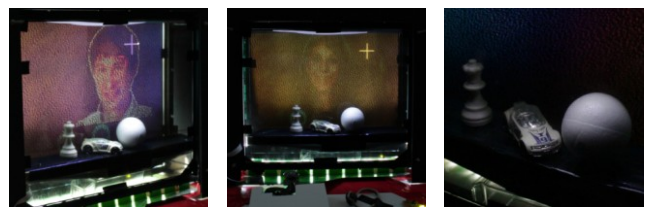


Figure 1. MUSTARD Prototype (center) with one of the augmented views (left) and the physical objects (right).

To make inroads into these challenges we present MUSTARD: a multi-user see-through augmented reality display as shown in Figure 1. MUSTARD allows users to inspect objects behind a glass panel while delivering view dependent information through the glass. The system consists of two liquid crystals (LC), as shown in Figure 3 (b), which are separated by a short distance within which physical objects can be placed. We use the front LC as the data-panel where the augmenting information is displayed whereas the back LC serves as a dynamic mask-panel that deals with enabling the multiple viewers' capability. We also identify several ways in which view conflicts, resulting from two or more views being displayed at the same

location on the data-panel, can be managed. To identify best design strategies we built a prototype system and tested various cross-talk functions. We finally report on a user study that examines the usability and effectiveness of the system from a users' perspective.

The main contributions of this paper are:

1. Design and implementation of a multi-user see-through display which is *not* based on head-worn displays.
2. Using a sandwich of liquid-crystals to create a dynamic random hole display (RHD) which provides better image quality through improved coverage of the entire data-panel.
3. Implementation of a range of crosstalk algorithms for the determination of the color value of conflict pixels.
4. A systematic evaluation of the crosstalk functions using image-quality metrics such as HDR-VDP2 and PSNR followed by a user-study showing that users are able to see and recognize images; demonstrating the system works.

RELATED WORK

The most common see-through display nowadays has to be worn. Such embodiments have been around since the 1960's in the form of optical (and video) see-through head-mounted displays (HMD) to support mixed-reality applications such as medical visualization, maintenance and repair, and robot path planning. An alternative compromise recently available is to force real object's appearance through small displays and cameras as present in hand held tablets or mobiles.

Both the above are suitable from an egocentric model driven augmentation of physical objects. However, there are many situations where users expect to be able to walk-up and engage with the physical environment without having to adorn themselves with extra devices. We are interested in supporting such walk-up and use scenarios.

DigiScope [3] is an early implementation of a system that uses a transparent see-through display to allow users to see information and physical objects through a transparent screen. A typical approach to this form of projection-based systems is to use a translucent projection film. The film reflects or transmits perpendicular to the plane of the glass any incident light coming for a specific acute angle. This allows users to observe both the projected digital data and the physical object behind the glass-pane to which the projection film is attached. A direct extension to this to support stereoscopic see-through displays has been demonstrated with ASTOR [14] that uses HOEs (Holographic Optical Elements) on the glass pane to direct different light sources to each eye.

See-through displays are not limited to displaying information only. Wilson [20], Hirakawa et al. [4] and Hirsch et al. [5] propose different interaction techniques to allow pointing at the screen or the physical object. WaveWindow [16] is another system that supports gesture based interactions with see-through displays. However, the

different solutions proposed so far allow a single user to view and use the system at a time. There is need to explore how see-through displays can be extended to allow multiple users to use the same display space at the same time.

Without the see-through capability, multi-view display systems have been demonstrated to work using different principles. One method is to use parallax strip barriers to occlude certain parts of the screen from one eye while allowing another eye to see them. Systems like Perlin et al.'s autostereoscopic display [15], Varrier [19], and Dynallax [17] demonstrate this concept. Lenticular barriers unlike parallax barriers are not limited to two views. Kooima et al.'s work [8], the MERL display [11], RCL3D [12] all of which are based on lenticular barriers allow more than two unique views. Another approach based on directional diffusion films is demonstrated with Lumisight [10]. Sakurai et al.'s work [18] using polarized lighting is another technique used to present multiple views to users.

Extending such displays for see-through is fraught with difficulties; use of lenticular barriers or diffusion films results in distortions in views of any object placed behind.

Of particular interest to us is the Illusion Hole [7] which moves away from traditional linear parallax barriers and uses a single circular viewport per viewer. Naschel et al.'s RHD prototype [13] extends their approach by using a large number of micro viewports and demonstrates its advantages over parallax barrier displays. Gu et al. [21] demonstrates a full multi-user multi-view system using this concept with their Tabletop Autostereoscopic Display (TAD). All these approaches to multi-view displays are based on viewing the data through a hole-mask that is placed at a certain distance from the data to serve as a barrier that mediates the view for different users.

While these approaches also break when we place an object between the data and the hole, it is possible to adapt them to create a multi-user see-through display. We present MUSTARD which overcomes the challenges of simultaneous multiple-user and multi-view augmentation for objects in a cabinet.

MUSTARD: PRINCIPLE AND DESIGN

MUSTARD allows users to inspect objects behind a glass panel while projecting view dependent information through the glass. The system consists of two liquid-crystal elements: a) a dynamic *hole-mask* that restricts the amount of data visible to each user; b) a *data-panel* where data for multiple views is displayed simultaneously. Data shown on the data-panel can be seen from a specific view only when light passes through the hole-mask. This procedure generates location specific views, as only parts of the data panel are visible through the mask for each user.

This principle of MUSTARD is derived from the IllusionHole [7] and TAD [21]. However there are four key distinctions:

- **Mask Type:** Previous approaches use a static hole-mask. A static hole-mask like in IllusionHole [7] and TAD [21] blocks parts of the data plane from being viewed by a user. In MUSTARD we used a dynamic random hole-mask allowing coverage of the entire screen by constantly changing the hole-mask from frame to frame.
- **See-through:** MUSTARD allows uninterrupted viewing of a physical object placed between the data-panel and the hole-mask. This object can be separately lit by non-polarized light with the only restriction of not saturating the light used for showing the data on the data-panel.
- **Ordering:** While previous approaches place the mask between the user and the data, we place the data-panel between the user and the mask allowing us greater flexibility in managing the views generated.
- **View Conflict Management:** View conflict results from two or more views being displayed at the same display location (pixels) on the data-panel. Through a careful and systematic analysis of various conflict management strategies we identify approaches to manage conflicts.

We now explain the design choices we made for arriving at our implementation of MUSTARD.

Hole-mask with see-through capability

When designing a hole-mask one needs to consider the *mask type* and the *ordering* of the data-panel and hole-mask with respect to the viewers. The mask type influences visibility of the object and the type along with ordering can affect see-through capabilities.

Static v/s Dynamic Mask

As mentioned before, IllusionHole [7] and TAD [21] rely on the ability of the system to block visibility of certain parts of the data panel to present view-specific information. Placing a large panel - the hole-mask, at a certain distance from the data-panel, does the blocking action.

A side-effect of a static hole-mask is that only a small part of the screen is visible to any user at any time resulting in reduced display coverage. An alternative approach to a static hole-mask is a controllable and dynamic hole-mask. Such dynamic hole-mask would allow its pattern to change over time in order to show the entire display screen to any user over a short period of time.

Conflict Management: The other advantage of a dynamic mask is related to the management of conflict pixels. Conflict pixels are regions of the data-panel that are simultaneously visible to more than one viewer. The data shown in these regions is visible to more than one viewer and leads to crosstalk for one or more users.

For a static mask (and with stationary users) the regions of conflict are fixed and the effect of crosstalk is more pronounced. While it is possible to minimize the effect of crosstalk in conflict pixels for a fixed mask [21], increasing the number of users increases the collision pixel count and

gradually degrades the image quality. For a dynamic mask, due to change in the position of the hole-mask the position of the conflict pixels changes over time. This change reduces the crosstalk at a single location and spreads it over the entire display screen.

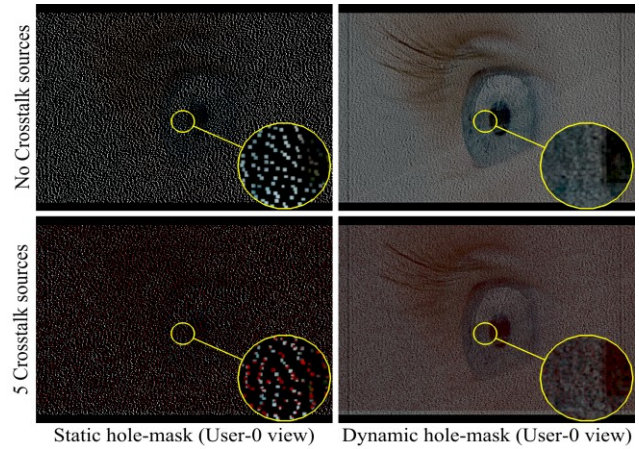


Figure 2. Effect of coverage & crosstalk on image quality with static & dynamic masks. Reddened pixels show crosstalk locations. Bright red indicates more crosstalk sources. Dynamic mask views are averaged over $\frac{1}{2}$ s or 30 frames.

Comparison of Mask Types: A comparison between a static mask and dynamic mask is shown in Figure 2. The static mask (Figure 2-Left) demonstrates a lower coverage. The black areas are regions where the view is blocked by the hole-mask. In case of the dynamic mask (Figure 2-Right), the view coverage is larger. The lower half of Figure 2 shows the effect of crosstalk sources. The red dots represent the location of the conflict pixels. A higher number of crosstalk sources leads to a more pronounced red dot. With the static mask, the locations of conflict pixels are constant. In case of the dynamic mask, the conflict pixels are distributed over the whole display. This minimizes the crosstalk effect at any single location.

Implementation Issues: A static hole-mask can be implemented as a printed mask on a transparent media. Dynamic hole-mask requires an active element. LC panels inside commonly available LCD screens can act as the active element. The linear polarizer inside an LCD screen is designed to attenuate light not polarized in the same direction as the polarizer. An LC panel is sandwiched between two crossed polarizers with polarization directions at 90° to each other as shown in Figure 3 below. The LC generates a twisting action on polarized light thereby controlling its passage through the front polarizer. This capability of LC panels can be used to generate a dynamic hole-mask by displaying a black and white dot pattern.

See-through and Panel Ordering

A simple implementation using two LC panels can work as a multi-user display. The hole-mask LC panel consists of an unmodified LCD screen with its backlight removed. This is placed in front of another LCD screen which serves as the

data-panel. If the polarization directions of the front polarizer of the data-panel LC to the rear polarizer of the hole-mask LC panel are aligned, then the system works as a multi-user display. When a white spot is displayed on the hole-mask LC, that region of the LC allows all the light from the data-panel to pass through to a viewer.

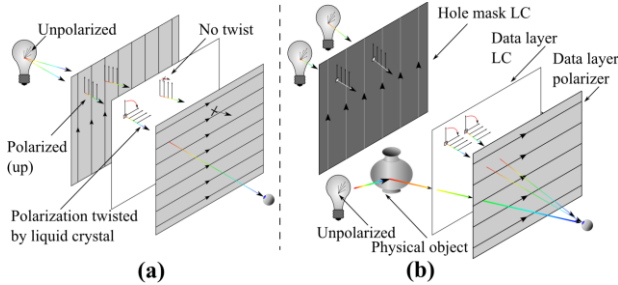


Figure 3. Operating principle for MUSTARD (a) Normal LC operation (b) Two LC setup with polarized light for data and unpolarized light for see-through aspect of the display.

The hole-mask LC will polarize all incident light passing through its rear polarizer. Hence such a naïve implementation cannot serve as a see-through display. To be see-through, the device needs to distinguish between light related to the displayed information and reflected light from the physical object. The hole-mask either allows the light related to the displayed information to pass through (a hole) or blocks it (mask). At the same time, the hole-mask allows all light from the physical object to pass through.

It is possible to achieve see-through capability by modifying an LCD screen. By design, if the rear polarizer is absent the remaining LC and front polarizer will allow unpolarized light to pass through it (except for a tiny fraction that is polarized 90°). It will also actively block any incident light that is polarized 90° to the front polarizer.

Anything displayed on this modified panel is not visible unless illuminated by correctly polarized light. Now if we were to display the information on such a panel and then illuminate it from behind with a polarized light source, a viewer should be able to see the information. At the same time, an object placed behind such a modified panel and illuminated by an unpolarized light source will be visible through the panel all the time.

This arrangement (see Figure 3(b)) is used to achieve see-through capability. The key difference of the order of the hole-mask and the data-panel now becomes apparent. The data-panel is in front of the hole-mask and the physical object is placed in between them. This arrangement converts the hole-mask into a grid of randomly distributed point-light sources. These holes emit polarized light necessary to view information on the data-panel. Also, each one of these holes illuminates any point on the data layer lying directly in line of sight to a viewer. As shown in Figure 3 above, two holes on the mask allow two points on the data layer to become visible to the viewer.

Conflict Management: With this arrangement, a single hole lies in direct line of sight of all users. But based on the user positions, the corresponding points on the data-panel are different. However, with a grid of holes, a single point on the display-panel may be illuminated by different holes for different users. So, the existence of conflict pixels is not eliminated and still needs to be managed. The process of generation of the hole-mask pattern and the final data-layer view are described next along with methods for conflict resolution for conflict pixels.

Mask and View Rendering

MUSTARD is designed to deliver different views to different users by using the same data-panel for display purpose. The rendering process is shown in Figure 4 below. The hole-mask illuminates only a part of the data-panel as seen from any one user's perspective. Thus only that part of the data-panel is visible to the user. This is referred to as the per-user data view. The data-panel displays a composite image which is generated from the combination of all the per-user data views. Per-user data views are generated using the user's eye-positions and the hole-mask pattern.

Mask Rendering: The hole-mask is an image with a collection of black-and-white dots. The white dots act as holes while the black dots act as masks. The locations of the white dots need to change over time so that over *10-frames* the white dots cover the entire display area. The ideal algorithm to compute this pattern is the Fast Poisson Disc Noise algorithm proposed by Bridson [1]. It has been used for static hole-masks by previous implementations [13,21]. This algorithm outputs a pattern of points at random locations but with uniform density. Since in our case the hole-mask is dynamic, a new pattern needs to be computed for every frame. Details of how this is done are given in the Algorithm Implementation section.

Generating Composite View Image: To generate data-panel's composite image the perspective-corrected view for each viewer has to be constructed using the eye-positions of each viewer. This step assumes that the entire view will be seen by the viewer. Then the per-viewer data view is processed against the hole-mask pattern to determine the visible parts of the view. This involves finding the point on the mask which is collinear to an eye position and a point on the data layer. If the point on the hole-mask pattern is a hole then the color data from per-viewer data view is retained for further processing. This process is carried out for all active users.

Conflict Management: The next step is to determine if there is a conflict at any point on the composite view by checking if more than one per-viewer data view has color data at the same position. In the absence of conflict, a single per-viewer data view contributes its value to the composite view. In case no data view contributes color, the output is black. Due to lack of a hole directly in line of sight from that point to any active viewer, the point is not of any interest for display purposes.

The final step is to resolve the conflicted pixels by using one of the functions (described next) to determine the color of the conflict pixel. Once the conflict is resolved, the output color is inserted at the conflict pixel location in the data layer. Finally both the hole-mask pattern and the data-panel views are displayed on the respective screens.

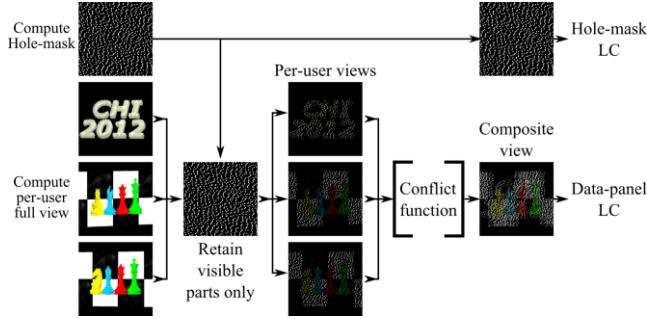


Figure 4. Steps involved in mask and view rendering.

Conflict Resolution Functions

MUSTARD's dynamic mask ensures that the conflict pixel locations are not constant from frame to frame. However resolving the color output for the conflict pixels is a critical part of the data display process. While TAD uses an error diffusion technique that spreads the error over neighboring pixels, we chose to apply the conflict resolution on the conflict pixel itself. The error for any single view is a function of the distance between the output color value and view's color value. The objective of the conflict resolution function is to choose the value that minimizes the sum of the absolute errors for all the contributing views.

At the pixel location where there is a conflict we convert the color values of each contributing view using a conflict resolution function (described below) and selecting the median of those contributing values.

The geometric median for a set of n points in the Euclidean space represents the point whose sum of distances from the n points is the least. This point also described as the 1-median provides the central tendency for the n points representing the user view colors. Computing the geometric median in two or more dimensions is a costly iterative process [2]. We use a simpler approach by processing color values over separate 1-dimensional channels. Our median function generates the median value from an $n \times 1$ kernel where each of the n values is contributed by a user view.

The input to the median function is perhaps the principal conflict resolution function that can be varied and there are many possible ways in which this function can be implemented. Following we describe six such options.

Full Range Method (FR): This is the simplest approach for determining the color of the conflict pixel. The color values contributing to the conflict pixel are directly passed to the median function for processing as shown in Figure 5 below.

Non-intersecting $1/n$ ranges method (NIINR): This method relies on splitting the overall dynamic range output into equal ranges of size $1/n$ for each user view.

Intersecting half ranges method (IHR): For this method the output range is also divided into n ranges however these ranges overlap with an offset of $1/(2(n-1))$ units as illustrated in Figure 5 (C). Note that for a 2-user view setting, the output of NIINR and IHR methods is same.

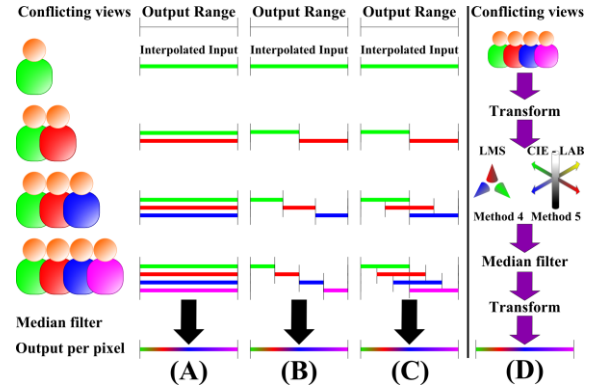


Figure 5. (A-D) Five of the six conflict resolution functions: (A) Full Range method, (B) Non-intersecting $1/n$ ranges method, (C) Intersecting half-ranges method, (D) LMS and CIE-LAB conversions. Each method is the input to the median filter that outputs the final value.

CIE-LAB and LMS Methods: The previous three methods operate on the RGB values and the median values are computed per channel. These methods are device-oriented. CIE-LAB and LMS methods operate in color spaces that are more suited for perceptual intent. The operation is similar to the FR method. The only difference is that the operations happen in CIE-LAB and LMS color spaces respectively. To achieve the conversions, the color values are converted to CIE-XYZ values and then to CIE-LAB and LMS. After applying the median function to the results, the output is obtained by re-conversion back to RGB values.

Alpha-Kern Method (AK): The Alpha-Kern, shown in Figure 6 below uses a scaled down version of the user view while retaining its original size. The downscaling process without resizing generates regions of no data which are filled with pixels from the original image at higher transparency. This generates an additional level for conflict source prioritization. The pixels belonging to the downscaled part of the view are given higher priority than the original pixels with higher transparency.

Thus if there are one or more user views which contribute color from their downscaled regions, the other sources with higher transparency are rejected from further processing. If there are no user views contributing color from their downscaled regions, all the higher transparency sources are considered for processing (while ignoring their transparency). The downscaling is achieved by a weighted average 3×3 kernel that colors the central pixel with the result of the weighted average.

MUSTARD IMPLEMENTATION

To evaluate the different conflict resolution functions we built a real prototype which is described next.

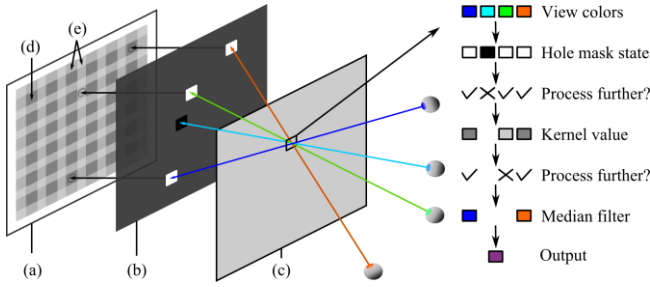


Figure 6. Alpha-Kern method. (a) Alpha mask, (b) Hole-mask pattern, (c) Data-panel. Output is the median of viewpoints indicated by blue and orange eye positions. Cyan eye view is rejected due to absence of hole. Green is rejected due to partial alpha contribution (e) in presence of (d) full alpha (blue and orange).

Physical Implementation

Our implementation consists of two 17" (1280x1024px) Matsushita LC panels extracted from existing PC monitors to simplify interfacing with a computer. We removed the rear polarizer from the front LC. The backlight setup consists of ten (30W, 6400K, 1535 lumens) compact fluorescent lamps behind a lenticular lens array (A4, 40 lpi). This provides a uniform diffused backlight.

The rear hole-mask LC and the front data-panel LC are 70 mm apart so as to accommodate a physical object within a 420 cm³ space while maintaining an object-to-display area ratio of 0.062:1. Any variation in this distance requires a one-time calibration necessary to convert the physical distance into coordinates used by the algorithm described next. We did not track the user's eyes as it can be achieved with existing technology and was not necessary to demonstrate the prowess our system.

Algorithm Implementation:

Coordinate Space Conversion

An Intel Core 2 Duo CPU and NVIDIA GeForce 9500 GT GPU were used. We selected the display area dimensions such that the texture coordinates of textures could be mapped 1:1 to the pixels of the displays. Using a one-time calibration the physical distance between the hole-mask LC and the data-panel LC is converted to coordinates in the texture-space coordinate system. Similarly, the user eye coordinates are mapped to texture-space coordinates. For the repeatability of the experiment across different conflict functions, the eye coordinates were held fixed. This was done by marking the physical locations of the viewer and not changing them for the duration of the experiment.

Cg Fragment Shaders

The steps for conflict pixel resolution are carried out at the GPU level per frame. The data-panel information is processed as a texture and all operations are carried out on every texture coordinate. The perspective corrected views

for each eye are drawn into FBO textures. The textures are processed as per the hole-mask pattern (another texture). The resulting output is passed through the conflict resolution function and the output is the final texture which is drawn to the data-panel LC. A single Cg-based fragment shader program performs all these actions.

Poisson Disc Noise for Hole-mask

In our implementation the hole-mask pattern consists of a black-and-white checkered pattern with a block size of 3x3 pixels. The location of the blocks in the pattern is computed first using the Poisson Disc Noise function. GPU based computation of the Poisson Disc Noise function is not optimal. Also, the dynamic pattern requirement doesn't preclude the reuse of a previous pattern as long as it is not reused in the very next frame. Thus, by pre-computing a set of m patterns the CPU can provide a list of patterns for the GPU to use. This frees the CPU to refresh the m patterns one at a time. Our implementation of Bridson's algorithm [1] ($r=5$ and $k=32$) gives us a hole coverage between 13 to 14% of the hole-mask per frame. The calculation time per hole-mask pattern is 10ms. We start with a pre-computed set of 4 masks and refresh one every 12 frames. Thus, the users see 8 different hole-masks every second at 60 fps.

Conflict Function Implementation

Processing conflict functions in the CIE-LAB and LMS color spaces requires conversion. To achieve this, the existing RGB values are converted to CIE-XYZ and then to the respective color spaces. All conversions use the formulae specified for Observer = 2° and Illuminant = D65.

Since the user views are dynamic and only a small part of each view is seen at a time, it is not practical to perform down-sampling on the entire view in the Alpha-Kern method. Instead, a fixed pre-computed alpha mask is used. The alpha mask consists of 3x3 pixels with weighted values of 16/256 for corners, 32/256 for edges and 64/256 for center. Once the position on the hole-mask pattern is identified to be a hole, the corresponding alpha value (at the same position) is retrieved from the alpha mask. If the value is greater than 32/256, the position is the central position in the 3x3 grid. The weighted average of the user view is calculated at the corresponding point on the data layer and its value is returned for further processing. If the position is not central, the returned color value is the original color value multiplied by the alpha value of the point. Prioritization based on the alpha values is performed and the final output is the median value of the prioritized inputs.

STUDY 1: EVALUATION OF CONFLICT FUNCTIONS

To examine the quality of output from a conflict function we compare the output in presence of crosstalk (test-source) with the expected output in the absence of crosstalk (reference-source). The reference-sources were derived from three different usage scenarios.

High-conflict Scenario: This scenario assumes that one user is presented with extreme conflict. This happens when all

the other users are viewing nearly similar information but very different to the view of the first user. The user views are shown in Figure 7. The first user view (Figure 7-left) has large areas of black space and the text has very large regions of white. This is contrasted against the chess piece set which is the scene for all the crosstalk source views (see Figure 7-center). Since all crosstalk sources have near similar contents, the crosstalk effect should be additively adverse to the first user view. This is apparent in the regions which are not black in Figure 7-right.

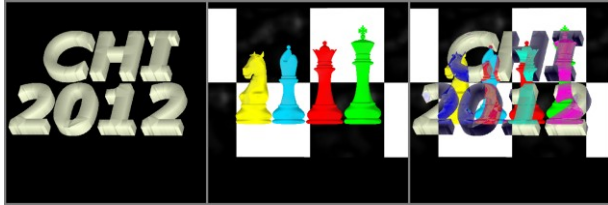


Figure 7. High-conflict scenario for conflict function evaluation. *Left:* First user view showing ‘CHI 2012’, also the view of interest. *Center:* One of the crosstalk sources for generating conflict pixels. *Right:* XOR of the two images showing conflict regions and color. Black indicates no conflict.

Perspective Visualization Scenario: In this scenario all users see a different view albeit from the same 3D scene. Such a scenario would exist when all users are shown a virtual 3D object that looks slightly different depending on the user's location as seen in Figure 8-left.

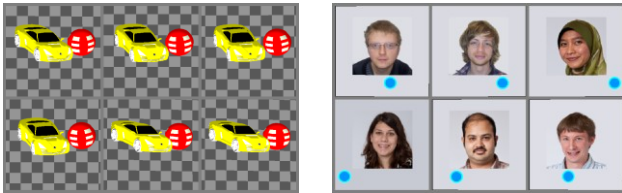


Figure 8. Other scenarios for evaluation of conflict functions. *Left:* 6 closely related views of the same 3D scene from different positions. *Right:* 6 views showing an example of possible personalized content.

Personalized Views Scenario: In this scenario all the users are viewing different content. For example, in a museum cabinet displaying a crown, and depending on their location, all the viewers can be shown a king's portrait each, from a chronologically ordered series of regents who have worn the crown. Figure 8-right shows this scenario.

Evaluation Technique

Using a camera to capture the view introduces an external factor to the visualization of the results. Hence we compared the conflict functions using screen-grabs of the outputs. Our first metric is the Peak Signal to Noise Ratio (PSNR) which is relevant due to the temporal nature of the views [6]. The second metric is Mantiuk's HDR-VDP-2 [9] which is a visual differences predictor that measures how much difference will be visible between our test source and reference source images. Specifically the quality mean opinion score (Q_{MOS}) allows us to compare the outputs by using the score for ranking.

One data-panel frame contains only 13-14% of a single per-user view. Also the dynamic mask is integral to uniform distribution of crosstalk which itself is a small part of the per-user view. Hence an image averaged over 10 frames is used for evaluation. Both metrics would output better scores for a single frame. The 10-frame average allows us to avoid this positive bias.



Figure 9. Sample of test source images. Top row: Views with two crosstalk sources. Bottom row: Views with four crosstalk sources. Methods shown (left to right): FR, CIE-LAB, AK.

Procedure

We used the algorithm implemented on our MUSTARD prototype for generating the data to be evaluated. To standardize the conditions, the user eye positions were considered fixed for the duration of the evaluation. The algorithm generated images meant for the first user view with (test source) and without crosstalk (reference source).

Screen grabs were then averaged over 10 frames for both sources. A sample set of images for the high-conflict scenario is shown in Figure 9. For PSNR we used the implementation available on Max-Planck Institute's website (<http://driiqm.mpi-inf.mpg.de/generator.php>). The Q_{MOS} metric was generated using the original reference implementation.

Results

PSNR Analysis

With PSNR a higher value is an indicator of better quality. The left column of Figure 10 shows the PSNR values for each of the test scenarios.

The average PSNR value is the lowest for the high-conflict scenario (125.59 dB), followed by perspective visualization scenario (133.23 dB) and personalized views scenario (138.16 dB). The high-conflict scenario was designed to demonstrate the ability of the system to generate viewable images even in the most adverse conditions. From Figure 9 we can see that the images are recognizable even when there are up to 4 strong crosstalk sources conflicting with the current view. As expected the PSNR value for each image drops as the number of crosstalk sources increases.

Our results also show that AK method has the highest PSNR values of all conflict management functions for each of the three scenarios. The AK method results in PSNR values of 128.36 dB, 140.38 dB and 143.88 dB for the three scenarios which is higher than the PSNR values of all other

techniques. It is also visible in Figure 9 that the images on the right column (corresponding to AK method) have better quality than the other images. The PSNR value for each of the scenario also suggests that CIE-LABS and CIE-LMS are almost identical in performance. It is also worth nothing that NI1NR method performs the worst in terms of PSNR.

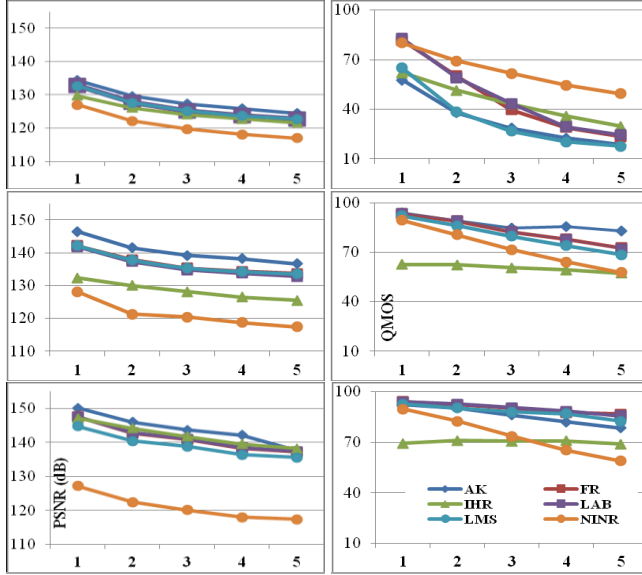


Figure 10. PSNR and Q_{MOS} results. Top: High-conflict scenario, middle: Perspective visualization scenario, bottom: Personalized views scenario. X-axis: Crosstalk source count.

Q_{MOS}

The Quality mean opinion score (Q_{MOS}) is a score between 0 and 100 with 0 implying worst quality and 100 implying best quality. Like the PSNR results, average Q_{MOS} scores are much lower for the high-conflict scenario when compared with the other two scenarios.

From Figure 10 we can see that for Q_{MOS} there is no clear winner in terms of Conflict functions. In the high-conflict scenario NI1NR turns out to be better than the rest with AK and LMS seeming to be the worst techniques. However when looking at the two scenarios that are closer to what one might encounter in a real-world application it seems like NI1NR starts to degrade quickly as the number of conflict sources increases. Whereas the AK technique performs well even when there are 4 or 5 cross-talk sources.

Discussion

The results of this study show that there are no clear winners in terms of best suited conflict function. However our AK method frequently performs better than the other methods and seems more resilient to quality degradation due to more cross talk sources.

We also find that NI1NR and IHR generally performed worse than most other techniques. Although NI1NR performed better than the other techniques when looking at Q_{MOS} scores for the high-conflict scenario we feel that this is more an artifact than a reflection of the quality of the

technique (in the remaining 5 cases it was at the bottom). Based on the results of this study and on Figure 9 it seems that visual quality in these cases might be better aligned with PSNR values than with Q_{MOS} .

USER STUDY

In order to further understand whether users are able to look at both the physical object and the image on the data-panel we carried out a user study where participants had to look at both the object and the data-panel to complete the trial.

We carried out a 3x3x3 factor within-subjects experimental design with the number of crosstalk sources, the conflict functions and the viewport as the factors. We used FR, AK and CIE-LAB as the conflict functions with 2, 4 and 5 crosstalk sources.

The conflict functions were chosen based on the results of our previous study. NI1NR and IHR were discarded based on their performance while CIE-LMS was not included as it was similar in performance to CIE-LAB. Each participant had to perform the task from three distinct viewports (Left, Center and Right). The position of the viewports was fixed.

Task

The task was a symbol matching task. For each trial, the participant had to look at the see-through display through a viewport. One of six pre-determined symbols appeared at the top of the data-panel. The same six symbols appeared as annotations, one above each of six physical objects behind the data-panel. The physical objects were 25mm square cardboard cutouts (see Figure 11-right) with the numbers 1 to 6 printed on them. The participant was asked to identify the physical object above which the matching symbol appeared and then call out the number printed on the matched object. The experimenter entered the number in the system through a keyboard. The system then moved to the next trial. Depending on the experimental condition there were 2 or more crosstalk sources.

After each trial, the mapping of the symbols to the physical objects was changed for both the participant's view as well as the crosstalk source views. No two views had the same symbol order during a trial.

Participants and Setup

10 volunteers (4 female) aged 20-40 yrs participated in this study. None of them had any experience with a multi-view system or a see-through display. The participants were given time to acclimatize themselves to MUSTARD. The trials commenced after suitable explanation of the device and task. Each participant completed 5 repetitions for each combination of factors resulting in a total of 135 trials per participant ($3 \times 3 \times 3 \times 5 = 135$). Participants were free to rest at any point when not performing a task.

The system was pre-calibrated to work with each of the viewports so there was no need for the system to include an eye-tracker (which would also have introduced an

additional level of complexity in the data-analysis). The viewpoints were within 800-1000 mm of the system. None of these offered any advantage by way of placement.

Procedure

The order of presentation of the crosstalk sources was first balanced using Latin Squares then the order of presentation of the conflict functions was balanced to remove any order effects. The participants completed all trials at any one specific viewport before moving to the next viewport.

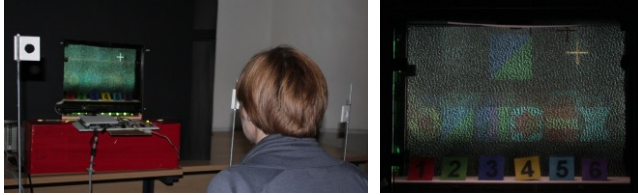


Figure 11. Left: Experimental setup. White squares with holes are the viewports. Right: View as seen by the user on the left. View has 2 crosstalk sources and CIE-LAB is the conflict resolution function in use. The view-dependent virtual annotations are shown above the real numbered objects.

Measures

The experimental program recorded whether the user identified the right object or not. If the user called out the wrong number then the system registered an error and the experiment progressed to the next trial. We also asked the users to tell us how confident they were of their answers on a scale of 0 to 2. '0' meant it's a pure guess and '2' meant they are extremely confident of their answer.

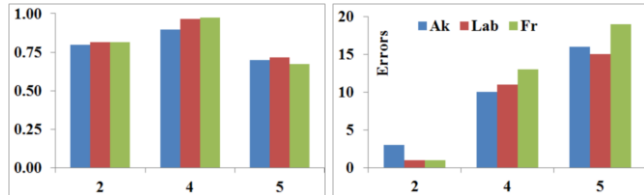


Figure 12. Left: Total Number of error across for each conflict function and crosstalk source. Right: Average weighted accuracy for each conflict function and crosstalk source.

Results

From the 10 participants we recorded a total of 1350 trials, of which only 89 trials resulted in an error giving an overall accuracy of 93%. As seen in Figure 12(Left) with the increase in the number of crosstalk sources the error rates increased significantly ($F_{2,18} = 57.4$, $p < 0.05$ – Univariate ANOVA with users as random factors). However we found no statistical difference in error rates between the different conflict resolution techniques. We created a weighted accuracy as $(\text{Accuracy} \times \text{Confidence score})/2$. This score reduces the weighting for trials where the users were not confident. The score was 0 if the answer was wrong or if the user had no confidence in their answer. Figure 12 (right) shows the average score of the weighted accuracy.

The results of this user study demonstrate that the users are able to successfully identify objects even in the high-

conflict scenario with up to 5 crosstalk sources. Users were able to successfully complete the task with the conflict functions tested further reiterating our claim that these functions are effective in reducing effect of crosstalk.

DISCUSSION

Hole-Mask

With MUSTARD, we demonstrated the concept of using two LCs to create a dynamic RHD which allows use of polarized and unpolarized light for achieving a multi-view see-through display. We also demonstrate that a dynamic mask based system provides better coverage over static mask. Below we discuss some of our observations.

Hole-mask Pattern

The size of the hole is governed by two extreme cases. The smallest size of the hole depends on the smallest pixel pitch of the hole-mask LC. The largest possible size of the hole is the one that doesn't allow any one data-panel pixel to be visible to two eyes at the same time. This size is given by the formula: $h_w = IPD * d_{LC} / d_{EYE}$. Here h_w is the hole width, IPD is the interpupillary distance (essentially the closest any two user eyes are assumed to be), d_{LC} is the gap between the data panel LC and the hole-mask LC and d_{EYE} is the distance of the user from the data panel LC.

Conflict Functions

We presented a set of conflict functions that can be used for MUSTARD. Our analysis of the methods based on image quality metrics showed that AK method performs better than others in certain conditions. However we found no difference between AK, CIE-LAB and FR in our user-study. These findings suggest that all these functions might be suitable as conflict function.

MUSTARD could be setup to select a conflict functions on the fly. The content to be delivered can determine the choice of function. Since our methods operate on a pixel level, individual pixels or regions can use entirely different functions. Thus the system could deliver a uniform user experience across the user group. Alternately, at the cost of quality of view for the group, a single user with higher priority could have access to the best view possible.

LC Selection

Hole-mask LC: The rear LC panel only serves as a white light source. A specialized LC panel with larger monochromatic pixels can be used so as to reduce the ringing effect caused by the fixed color filter coating on the LC. At the same time, it is necessary to resize the rear LC such that optimal coverage of all view positions is achieved.

Limitations: The LC gap and hole-size are limiting factors for MUSTARD as they affect the view distance. While the current implementation focuses on the effectiveness of conflict functions, we feel that the individual investigation of these factors could be undertaken as future work.

Physical Objects and User Experience

Objects Sandwiched Between Two LC Panels

With our prototype, the physical object is placed between the two LC panels. This arrangement is not a conceptual limitation. A wedge waveguide placed immediately behind the data layer could be an alternate implementation. The angle of input of the light fed to the bottom of the wedge determines the position of exit. The input to the wedge would be MUSTARD's hole-mask pattern (compensated for polarization shift due to reflection) and the conflict resolution methods described previously will still be valid for such an implementation.

User Experience:

The physical object is visible to any user and is not occluded by the displayed content. While there may be arguments in favor of overlaid object annotations, MUSTARD's multi-view approach prefers annotations around the object. This can be particularly important in the intended usage scenarios wherein a user in a group is interested in the physical object only. In this case MUSTARD delivers an unobstructed view to that particular user while presenting data to the other users.

CONCLUSION

We have demonstrated the concept of MUSTARD for achieving multi-user see-through effects suitable for walk-up and use scenarios. Through a two LC based prototype system we show the feasibility of a random hole display that includes a dynamic mask to improve image quality. We examine various functions to manage conflicts in pixel color due to multiple user views. We show that the alpha kernel conflict management functions provides good image quality scores as well as performing well in a user study.

ACKNOWLEDGEMENTS

We thank Dr. Tania Pouli for the help with the HDR-VDP and Benjamin Long for advice on Cg. This work is partially supported by Microsoft Research through the PhD Scholarship Programme & by a UoB Pump Priming grant.

REFERENCES

1. Bridson, R. Fast Poisson disk sampling in arbitrary dimensions. In SIGGRAPH 2007 Sketches.
2. Chandrasekaran, R. and Tamir, A. Open questions concerning Weiszfeld's algorithm for the Fermat-Weber location problem. Mathematical Programming (1989).
3. Ferscha, A. and Keller, M. DigiScope: An Invisible Worlds Window. In Ubicomp 2003 (2003), 261-264.
4. Hirakawa, M., Kojima, Y. and Yoshitaka, A. Transparent interface: a seamless media space integrating the real and virtual worlds. In Proc. Human Centric Computing Languages and Environments (2003). 120-122.
5. Hirsch, M., Lanman, D., Holtzman, H. and Raskar, R. BiDi screen: a thin, depth-sensing LCD for 3D interaction using light fields. In SIGGRAPH Asia 2009.
6. Huynh-Thu, Q. and Ghanbari, M. Scope of validity of PSNR in image/video quality assessment. Electronics Letters 44 (2008), 800-801.
7. Kitamura, Y., Konishi, T., Yamamoto, S. and Kishino, F. Interactive stereoscopic display for three or more users. In SIGGRAPH 2001, ACM (2001), 231-240.
8. Kooima, R., Prudhomme, A., Schulze, J., Sandin, D. and DeFanti, T. A multi-viewer tiled autostereoscopic virtual reality display. In Proc. VRST 2010, (2010).
9. Mantiuk, R., Kim, K. J., Rempel, A. G. and Heidrich, W. HDR-VDP-2: a calibrated visual metric for visibility and quality predictions in all luminance conditions. In SIGGRAPH 2011, ACM (2011), 1-14.
10. Matsushita, M., Iida, M., Ohguro, T., Shirai, Y., Kakehi, Y. and Naemura, T. Lumisight table: a face-to-face collaboration support system that optimizes direction of projected information to each stakeholder. In CSCW '04.
11. Matusik, W. and Pfister, H. 3D TV: a scalable system for real-time acquisition, transmission, and autostereoscopic display of dynamic scenes. In SIGGRAPH 2004, (2004).
12. Morishima, H., Nose, H., Taniguchi, N., Inoguchi, K. and Matsumura, S. Rear-cross-lenticular 3D display without eyeglasses. In Stereoscopic Displays and Virtual Reality Systems V, (1998).
13. Nashel, A. and Fuchs, H. Random Hole Display: A non-uniform barrier autostereoscopic display. In 3DTV Conference: The True Vision - Capture, Transmission and Display of 3D Video, (2009), 1-4.
14. Olwal, A., Lindfors, C., Gustafsson, J., Kjellberg, T. and Mattsson, L. ASTOR: An Autostereoscopic Optical See-through Augmented Reality System. In ISMAR '05.
15. Perlin, K., Paxia, S. and Kollin, J. S. An autostereoscopic display. In SIGGRAPH '00, (2000).
16. Perry, M., Beckett, S., O'Hara, K. and Subramanian, S. WaveWindow: public, performative gestural interaction. In ITS '10, ACM (2010), 109-112.
17. Peterka, T., Kooima, R. L., Sandin, D. J., Johnson, A., Leigh, J. and DeFanti, T. A. Advances in the Dynallax Solid-State Dynamic Parallax Barrier Autostereoscopic Visualization Display System. IEEE Transactions on Visualization and Computer Graphics 14, (2008).
18. Sakurai, S., Kitamura, Y., Subramanian, S. and Kishino, F. A visibility control system for collaborative digital table. Personal Ubiquitous Comput. 13, (2009).
19. Sandin, D. J., Margolis, T., Ge, J., Girado, J., Peterka, T. and DeFanti, T. A. The Varrier autostereoscopic virtual reality display. ACM Trans. Graph. 24, (2005), 894-903.
20. Wilson, A. D. TouchLight: an imaging touch screen and display for gesture-based interaction. In ICMI, (2004).
21. Ye, G., State, A. and Fuchs, H. A practical multi-viewer tabletop autostereoscopic display. In ISMAR '10 (2010).